ANALOG DEVICES

AN-776
APPLICATION NOTE

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106 • Tel: 781/329-4700 • Fax: 781/461-3113 • www.analog.com

Dual AD9981 Design Guideline to Achieve UXGA Resolutions

by Del Jones

INTRODUCTION

Pixel clock speeds in excess of 110 MHz can be achieved with the AD9981 by using a dual chip "ping-pong" configuration. A dual chip solution is different from an alternate pixel sampling solution in that full refresh rates can be maintained.

There are many ways to implement a dual AD9981 design. This application notes serves to make the user aware of the considerations that should be weighed when implementing this configuration. Among the variables are layout and routing constraints, clock selection, graphics controller requirements, and maximum speed requirements.

Analog Input Layout and Routing

When laying out and routing the analog inputs (R, G, B, and HSYNC), several factors should be considered. The trace lengths of the R, G, and B inputs should be kept as equal as possible, while also keeping the routes direct (no zigzagging) to maintain equal propagation delays. The branches to each of the AD9981's analog inputs should be kept as short as possible. The 75 Ω terminators on the RGB inputs should be placed as close to the branch junctions as possible. Lastly, each R, G, and B branch requires its own coupling capacitor. These considerations are illustrated in Figure 1.

Clock Source Selection

There are three methods that can be used for clocking data. An external clock source can be used to clock both AD9981s as well as the data latching device(s) (graphics controller). This method requires external PLL circuitry and special high speed clock layout and routing considerations.

A second option is to use the PLL in chip 1 to drive chip 2. This method would require chip 2 to be configured for external clock operation, using the negative edge of chip 1's DATACK to sample the RGB data. This method employs the most direct routing of HSYNC. The HSYNC could be routed directly to chip 1, and then routed to the second device. (Although the second device does not use HSYNC to generate a clock, it is still needed to provide a timing reference for other functions, such as clamping.) The problem with this option is that it causes difficulty in setting the correct clock phase of chip 2 because of the added propagation delay between HSYNC and chip 1's data clock output. It will also cause ongoing clock phase difficulty in chip 2 because of the variability of chip 1's data clock propagation delay over time and temperature.

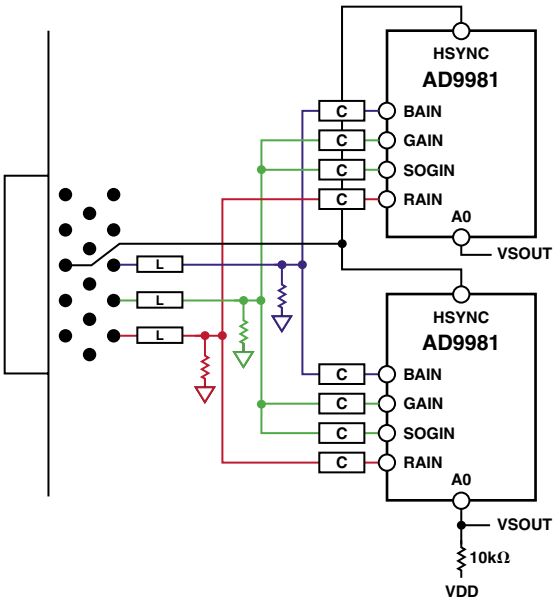

Figure 1. Analog Input Routing

The recommended method for clocking is to use the PLLs in both chips. This method requires special attention to HSYNC input layout, as shown in Figure 1. If very careful attention is paid to keeping the branch lengths identical (avoiding zigzagging), then the skew between the two chips' sampling clock and digital outputs will be negligible.

Sampling Clock Inversion

All three clocking methods described earlier require chip 2 to sample RGB data 180° out of phase with chip 1. This can be achieved by using the phase adjustment control. The initial phase setting of chip 2 would be offset 180° (nominally, 16 steps) from chip 1's initial phase setting. Using this method, the data output from chip 2 will be shifted by 1/2 pixel clock. This method allows both chips to run their clocks at half of the effective data rate. Chip 1 would capture odd data on its sampling edge, while chip 2 captures even data on its sampling edge (180° out of phase from chip 1).

The timing issues associated with data capture are negligible if the capture device has separate capture clocks for each data port (odd and even data). If only one clock pin is available on the latching device (graphics controller, ASIC, and so on) for both ports, it will be necessary to invert the clock internal to the latching device. This would be true unless the latching device has the ability to capture data on either edge of the data clock. Figure 2 illustrates the basic timing relation between the incoming data pixels and the digitized output data for both devices.

Clock Phase Adjustment

Although the internal clock delays should be the same, each chip's phase will need to be adjusted separately. Phase differences between the two chips can come from layout trace length variations in HSYNC or RGB inputs, and from normal internal chip variations.

Since each chip is running at half speed, the phase adjustment step sizes are doubled with respect to the full speed clock. This results in 1/2 the number of useable phase adjustment steps (16 instead of 32) since the phase adjustment range will now cover two full speed pixels rather than one.

It is recommended that the phase adjustment for the two devices be performed independently since the optimal phase setting for each device may not differ by exactly 16 phase steps (for various reasons). To speed the phase selection algorithm, a limited phase selection process could be used on device 2 after the full process is performed on device 1. If the optimal phase setting (OPS1) is less than 16 on device 1, then it is reasonable to expect that the optimal phase setting for device 2 (OPS2) is OPS1 + 16 ±4 (wrap around if result is > 31). If OPS1 is ≥ 16 then OPS2 = OPS1 – 16 ±4 (wrap around if result is < 0). Therefore, the second step of the phase selection algorithm (for device 2) can be limited to those steps that are expected to be the optimal setting.

Interpart Difference Adjustments

Dual ADC applications are sensitive to the differences between the two ADCs. These differences can come from gain, offset, and linearity.

Gain and Offset

Dual ADC applications are highly sensitive to gain and offset errors between the two chips. Any difference between odd and even pixels is highly visible. *Therefore, accurate gain and offset adjustment is required for each chip*. Fortunately, the AD9981 has the automatic clamp feedback feature. When enabled, the clamp feedback automatically eliminates the channel-to-channel differences in offset for all channels.

While the differences in channel-to-channel gain will not affect the image as much as differences in offset, it is still recommended that any channel-to-channel gain mismatch be minimized using a one-time factory calibration scheme.

Linearity/Dithering

Dual ADC applications are also sensitive to differences in linearity between the two devices. The use of 10-bit ADCs greatly improves the linearity performance of a dual ADC application so that dithering may not be necessary. However, if the application's requirements prove to be stringent, dithering can be used to enhance the linearity even further. The dithering method requires that on every other data frame the even and odd device be swapped. For example, during frame one, device 1 processes odd pixels and device 2 processes even pixels. During frame two, device 1 processes even pixels while device 2 processes odd pixels. *Dithering allows the eye to essentially "average" the effect of the linearity differences as well as differences in offset and gain*.

Device Addressing

Each AD9981 requires a different serial bus address. This is achieved using the A0/VSOUT pin, shown in Figure 1.

Data Mode Selection

With dual port devices such as the AD9884A and the AD9888, it was possible to operate in three different data modes when implementing a ping-pong scheme. However, since the AD9981 is a single-port design (only 30 output bits), single-channel mode is the only mode a dual AD9981 design can operate. See Figure 2.

Single-channel mode requires 60 data output traces and requires a graphics controller that has 60 data input lines. It is recommended to run in single-chip mode (device 1 processing every pixel and device 2 in low power mode) for frequencies up to 110 MHz and in ping-pong mode for frequencies over 110 MHz. Data switching limits the maximum operating speed to 220 MHz.

In order to achieve the proper DATACK frequency (Pixel Clock ÷ 2), it is necessary to program the PLLDIV registers appropriately. When operating in ping-pong mode, PLLDIV = Pixel Frequency ÷ (2 × Horizontal Frequency). This is 1/2 the normal value for PLLDIV.

Reference Designs

Ping-pong reference designs using the AD9981 are not currently available. For more information, please contact the factory via email at flatpanel_apps@analog.com.

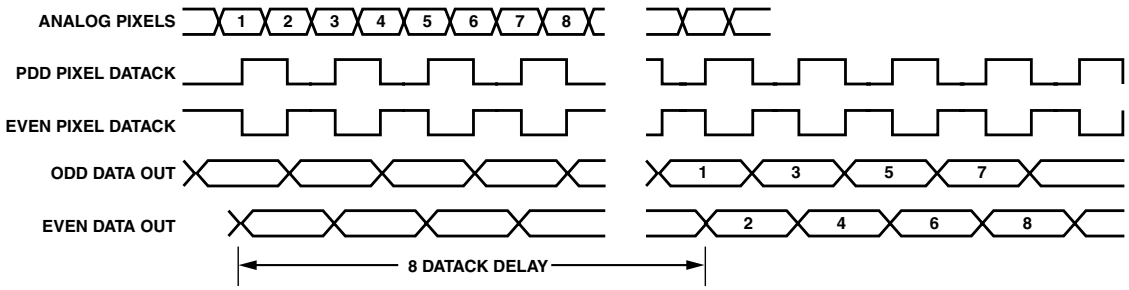

Figure 2. Basic Data Timing for Dual AD9981 Implementation

AN05438–0–3/05(0)